SYSTEM AND METHOD FOR DOWNLOADING APPLICATION COMPONENTS TO A CHIPCARD

Technical Field

The present invention describes a system and method for downloading application components via distributed systems to chipcards, in particular to chipcards which are already in use.

Background of the Invention

Normally chipcards are shipped with prepared on-card application components.

These on-card application components permit communication between the chipcard and the chipcard applications, the so-called off-card applications, which are installed on a terminal, e.g. a server system. The chipcard - i.e. the on-card application component - communicates via a chipcard reader with this off-card application. Modern chipcards, so-called multifunction chipcards such as Java Cards or Smart Cards for Windows, have additional functionality permitting on-card application components to be mounted on the chipcard retrospectively, i.e. after the chipcard has been shipped (see FIG. 1). In such cases the on-card application components are downloaded from the terminal to the chipcard via the chipcard reader.

VISA, for example, has defined an Open Platform Specification describing the commands between the off-card

application and the on-card application component, the on-card interface and the security standards. OCF (Open Card Framework) and Microsoft's PC/SC on the other side address the communication between the application, the chipcard reader and the chipcard.

The more widespread use of distributed systems has resulted in an increasing need for downloading of on-card application components to the chipcard via distributed systems. The risks of such methods are obvious. The network is subject to varying loads, so the download may take a long time depending on capacity. Another key aspect in this context is security. All data transfers from the server via the client to the chipcard must be safeguarded. It must be ensured that a simple, secure authentication and encryption method which responds to the varying loads on the network is used when downloading application components.

At present, however, no systems or methods are believed to address this possibility.

Summary of the Invention

It is therefore the object of the present invention to deliver a system and method for downloading application components via distributed systems to a chipcard in a simple manner, taking account of the necessary security checks.

This object is fulfilled by the characteristics of Claims 1, 17, 18 and 20. Advantageous embodiments of the present invention are presented in the sub-claims.

The advantages of the present invention lie in the fact that downloading of the application components is divided into two stages.

The first stage occurs on the server only, and ensures that not every command to download the application components is sent individually over the network. This is effected by means of an optimized protocol which bundles the individual commands to download the application component into a command sequence and sends it as a data packet over the network. This reduces the time required for downloading application components over the network. Each command within the command sequence is assigned a digital signature and, where appropriate, encrypted. This ensures that only authenticated commands are accepted by the chipcard.

In this way this invention meets security requirements for the transfer of data via distributed systems, in particular the Internet.

The second stage occurs between the client and the chipcard, and ensures that the data packets are unpacked and sent individually to the chipcard.

All security-relevant keys and certificates are stored on the secure server. Communication between the client and

the server runs preferentially via SSL (Secure Sockets Layer) as the transfer protocol. Misuse of the inventive system/method is thereby rendered much more difficult.

Additional features and advantages are realized through the techniques of the present invention. Other embodiments and aspects of the invention are described in detail herein and are considered a part of the claimed invention.

Brief Description of the Drawings

The subject matter which is regarded as the invention is particularly pointed out and distinctly claimed in the claims at the conclusion of the specification. The foregoing and other objects, features, and advantages of the invention are apparent from the following detailed description taken in conjunction with the accompanying drawings in which:

FIG. 1 shows the state of the art of communication between the off-card application and on-card application component.

FIG. 2 shows a distributed communications architecture on which the present invention is based.

FIG. 3 shows the inventive steps involved in downloading on-card application components from a server over a network to a chipcard.

FIG. 4 shows the inventive architecture in accordance with FIG. 3 in a Java implementation.

FIG. 5 shows the inventive steps involved in downloading on-card application components from a server over a network to a chipcard in a Java implementation.

Best Mode for Carrying Out the Invention

FIG. 1 shows the state of the art in downloading of on-card application components from a terminal to the chipcard and in communication between the on-card application component and off-card application. In the state of the art the chipcard applications consist of an off-card application stored on a terminal and an on-card application component stored on the chipcard in the nonvolatile memory (see FIG.1). The terminal consists of a data processing unit with a chipcard reader and the corresponding driver software for the chipcard reader. The on-card application component communicates with the off-card application over several layers. Layer 1 defines the physical transfer protocol. Layer 2 superimposes that protocol with a logical, byte-oriented protocol. Layer 3 maps higher programming language on layer 2. An example of layer 1 is the protocol T=0, T=1 (ISO/IEC7816-3), layer 2 APDU

protocol (ISO/7816-4), layer 3 OCF (Open Card Framework) or PCSC ( ).

Normally the on-card application component is transferred to the chipcard via a loader application which runs on the terminal. In this process suitable chipcard commands are used (e.g. for file-oriented chipcards "CREATE" and "UPDATE" commands). At present no solution for the transfer of on-card application components via distributed systems to the chipcard is yet known.

FIG. 2 shows the inventive architecture of the present invention. The inventive architecture is based on a client/server architecture. The client communicates with the server over a network, e.g. the Internet or an Intranet. The client is connected to a chipcard reader and only the server has access to the secret keys required to download on-card application components to the chipcard. The keys may either be stored on the server itself or on another system to which the server has access. The chipcard is protected against unauthorized downloading of on-card application components in such a way that it only accepts commands when they are signed and/or encrypted with the correct keys. On the client a runtime program must exist which communicates both with the chipcard and with the server and which implements a protocol dependent on the respective chipcard.

This protocol specifies when which messages must be exchanged with the chipcard and the server. On the server a runtime program must exist which communicates with the

client and uses the keys accessible to the server as necessary, and which implements a protocol specifying when which messages must be exchanged with the client and when which keys must be used. The chipcards used are common chipcards (such as Java Cards or file-oriented chipcards) which do not have to be adapted for the present invention.

FIG. 3 shows the inventive steps for downloading of on-card application components from a server over a network to a chipcard.

The client establishes communication with the chipcard and with the server.

The client sends a request to the server for an on-card application component (application component A) to be placed on the chipcard. The client and server communicate preferentially via TCP/IP or HTTP.

The server sends a response to the client with the request to transmit the chipcard identification data and, where appropriate, a random number for authentication purposes. Chipcard identification data as a minimum contain data relating to the chipcard type and the chipcard number. The client receives the response from the server and sends appropriate command APPUs to the chipcard in order to retrieve the chipcard identification data and, where appropriate, a random number. The chipcard identification data are stored in the nonvolatile memory of the chipcard and can be read by means of suitable commands. The chipcard

receives the commands and returns the chipcard identification data and, where appropriate, the random number to the client. The client sends these data in a request to the server.

The server receives the request and evaluates the chipcard identification data to find out which keys have to be used, or to derive the necessary keys from Master Keys, in order to be able to download the application component A. The keys are used to prepare a command sequence for downloading of the application A from the server to the chipcard. This command sequence causes the application A to be created on the chipcard. The command sequence is a predefined sequence stored in the nonvolatile memory area of the server for a specific application. A further embodiment of the invention is that the command sequence is created in whole or in part with the aid of a program on the server. This is preferentially applied where card-specific data are also to be integrated into the on-card application component by means of the command sequence. Preferentially each command within the sequence is signed with the aid of the key (Session Keys) and encrypted as necessary. This can be effected, for example, by assigning the first command within the sequence a MAC (message authentication code) with the aid of the random number and the correct key, and assigning all subsequent commands a MAC based on the MAC of the preceding command and the correct key. The sequence with the signed and, where appropriate, encrypted commands is sent to the client.

The client receives the response with the command sequence and sends the commands consecutively to the chipcard. The chipcard checks the signature and only executes the commands if the signature is correct.

FIG. 4 shows the inventive architecture in accordance with FIG. 3 in a Java implementation.

On the client a Web Browser is run to enable the user to navigate to the Web page of the server. The Web page of the server contains the applet which implements the client program described in FIG. 3. When the Web page is displayed the applet is downloaded from the server to the Browser. The applet establishes a communication link to a servlet on the server. The servlet has the functionality of the server program.

The procedure for downloading the on-card application component corresponds to that set out in FIG. 3.

FIG. 5 shows the inventive steps for downloading of on-card application components from a server over a network to a chipcard in a Java implementation.

It is assumed in this that a brokerage application stored on a server is to be loaded into the chipcard. Authentication keys are also stored on the server.

The client establishes communication with the chipcard and with the server. Communication with the chipcard is implemented by OCF (Open Card Framework).

The client sends a request to the server for the brokerage application (on-card application component) to be placed on the chipcard. The client and server communicate preferentially via TCP/IP or HTTP.

The server sends a response to the client with the request to transmit the chipcard identification data (GetCardInfo).

The client receives the response from the server and sends appropriate command APPUs to the chipcard in order to retrieve the chipcard identification data. The chipcard identification data are stored in the nonvolatile memory of the chipcard and can be read by means of suitable commands. The chipcard receives the commands and returns the chipcard identification data to the client. The client sends these data in a request to the server.

The server receives the request and evaluates the chipcard identification data to find out the card type. An authentication method is chosen depending on the card type. In the present implementation the card type is a VISA Open Platform card with symmetrical keys. The first authentication step involves the server generating a random number and selecting a key number, and then sending that information packed in a command to the client. The client

extracts the OCF command and sends it to the OCF interface on the client computer. The OCF interface converts the OCF command into one or more APDUs and sends it/them to the chipcard. The chipcard receives the APDUs, identifies them as an authentication command, generates a random number, creates a Session Key from the two random numbers and the transmitted key, and thereby returns the random numbers in encrypted form.

The client transmits the card's response to the server. The server likewise generates a Session Key from the two random numbers and the key number. With the aid of this Session Key it checks the encrypted random numbers. If the check is successful the card is classed as authenticated.

The server sends a second authentication command to the client in order to authenticate itself according to the same method, as already described. If the check is successful the server is classed as authenticated.

The brokerage application is signed on the server by means of the Session Keys and encrypted as necessary in order to be able to download the broker application. This command sequence causes the application A to be created on the chipcard. The command sequence is a predefined sequence stored in the nonvolatile memory area of the server. A further embodiment of the invention is that the command sequence is created in whole or in part with the aid of a program on the server. This is preferentially applied where

card-specific data are also to be integrated into the on-card application component by means of the command sequence.

Preferentially each command within the sequence is signed with the aid of the key (Session Keys) and encrypted as necessary. This can be effected, for example, by assigning the first command within the sequence a MAC (message authentication code) with the aid of the random number and the correct key and assigning all subsequent commands a MAC based on the MAC of the preceding command and the correct key. The sequence with the signed and, where appropriate, encrypted commands is sent to the client.

The client receives the response with the command sequence and sends the commands consecutively to the chipcard. The chipcard checks the signature and only executes the commands if the signature is correct.

The steps outlined can also be used to customize the new application/brokerage application.

The present invention can be included in an article of manufacture (e.g., one or more computer program products) having, for instance, computer usable media. The media has embodied therein, for instance, computer readable program code means for providing and facilitating the capabilities of the present invention. The article of manufacture can be included as a part of a computer system or sold separately.

Additionally, at least one program storage device readable by a machine, tangibly embodying at least one program of instructions executable by the machine to perform the capabilities of the present invention can be provided.

The flow diagrams depicted herein are just examples. There may be many variations to these diagrams or the steps (or operations) described therein without departing from the spirit of the invention. For instance, the steps may be performed in a differing order, or steps may be added, deleted or modified. All of these variations are considered a part of the claimed invention.

Although preferred embodiments have been depicted and described in detail herein, it will be apparent to those skilled in the relevant art that various modifications, additions, substitutions and the like can be made without departing from the spirit of the invention and these are therefore considered to be within the scope of the invention as defined in the following claims.